
outbit Documentation

Release 0.0.7

David Whiteside

September 14, 2016

1	Overview	1
1.1	Why outbit?	1
1.2	Getting Started with outbit	1
1.3	Config Files	3
2	CLI	5
2.1	Actions	5
2.2	Inventory	5
2.3	Jobs	5
2.4	Logs	6
2.5	Plugins	6
2.6	Roles	7
2.7	Schedules	7
2.8	Secrets	8
2.9	Users	8
3	GUI	9
3.1	GUI	9
4	outbit	11
5	Installation	13
6	Usage	15
7	License	17
8	Author	19

Overview

1.1 Why outbit?

outbit provides a single UI for managing systems and allows system administrators the ability to hand off tasks they have automated to other individuals within an organization. outbit integrates with Ansible and adds RBAC, logging, secret keystore, and other features for using Ansible in a collaborative way.

1.2 Getting Started with outbit

Installing outbit client

```
$ pip install outbit
```

Installing outbit api server

```
$ pip install outbit
$ sudo outbit-api-install
```

Starting the outbit api server.

```
$ outbit-api -s 127.0.0.1 --insecure
```

Installing and Starting the outbit api server using Docker.

```
$ docker pull starboarder2001/outbit
$ docker run -d -p 8088:8088 starboarder2001/outbit
```

Login to the outbit shell. On the first login you will be prompted to change the default password. If your using the Docker container you can remove the “-insecure” flag since by its configured to use ssl. If you are using valid ssl certificates and not self signed certificates you can also remove the “-no-check-certificates” flag.

```
$ outbit -u superadmin -s 127.0.0.1 --insecure --no-check-certificates
Password: superadmin
Changing Password From Default
Enter New Password: *****
Enter New Password Again: *****
```

outit CLI Basics

The help command will display all the commands available to run.

```
outbit> help
actions list          list actions
actions del          del actions
actions edit         edit actions
actions add          add actions
users list           list users
users del            del users
users edit           edit users
users add            add users
roles list           list roles
roles del            del roles
roles edit           edit roles
roles add            add roles
secrets list         list secrets
secrets del          del secrets
secrets edit         edit secrets
secrets add          add secrets
secrets encryptpw    Change password encryption
plugins list         list plugins
ping                 verify connectivity
logs                 show the history log
help                 print usage
jobs list            list jobs
jobs status          get status of job
jobs kill            kill a job
schedules add        add schedule
schedules edit       edit schedule
schedules list       list schedules
schedules del        del schedule
inventory list       list inventory
inventory del        del inventory item
exit
```

The exit command will exit the application.

```
outbit> exit
```

The logs command will display the history of actions performed.

```
outbit> logs
```

outbit CLI Non-Interactive Usage

You can run commands with outbit from the bash shell without entering the interactive outbit shell.

```
$ outbit 'logs'
Password: *****
superadmin / ping None 06/18/2016 09:19
superadmin / ping None 06/18/2016 09:19
superadmin / help None 06/18/2016 09:19

$ outbit 'logs' 'users list'
Password: *****
superadmin / ping None 06/18/2016 09:19
superadmin / ping None 06/18/2016 09:19
superadmin
```

If you do not wish to type the password for each login attempt, you can set your password in the outbit configuration file.

1.3 Config Files

The search path for the outbit config files is your home directory (`~/outbit.conf`, `~/outbit-api.conf`) and then `/etc/` (`/etc/outbit.conf`, `/etc/outbit-api.conf`). The config file for the outbit CLI is `outbit.conf` and the API server is `outbit-api.conf`.

Below are the configuration options available on both the outbit CLI client and API server.

- `port` - Tcp port to connect (client) or listen on (server).
- `server` - IP address or hostname of server to connect to (client) or the server listen address (server).
- `secure` - Encrypt communication using https. This is to prevent passwords being transmitted clear text over the network.

Below are the configuraiton options only available for the outbit CLI client.

- `user` - User login name.
- `password` - User login password.
- `ssl_verify` - Verify the ssl certificate of the server is valid. This is to protect against man-in-the-middle attacks.

Below are the configuration options only available for the outbit API server.

- `ssl_key` - Private SSL key to use.
- `ssl_cert` - Public SSL key to use.
- `encryption_password` - Encrypt sensitive information in mongodb using this password. If you change this password after adding secrets, use `'secrets encryptpw oldpw=XXXX'` to migrate secrets to the new password.

Below shows example config files to use for reference.

```
$ cat ~/.outbit.conf
---
port: 8088
password: secretpassword
secure: False
ssl_verify: False

$ cat ~/.outbit-api.conf
---
port: 8088
server: 127.0.0.1
secure: False
encryption_password: "secretencryptpasswordForSecrets"

$ cat /etc/outbit.conf
---
port: 8088
server: 192.168.1.100
password: secretpassword
secure: True
ssl_verify: True

$ cat /etc/outbit-api.conf
---
port: 8088
server: 0.0.0.0
secure: True
encryption_password: "secretencryptpasswordForSecrets"
```


2.1 Actions

The actions command is used to create user defined actions. This is how the functionality of outbit is extended and customized for each environment.

```
outbit> actions add name=helloworld action=helloworld category=/testing command_run="echo 'hello worl
outbit> actions add name="ansibletest" action="ansibletest" category="/" desc="test ansible" pla
outbit> actions list
  name="helloworld" action="helloworld" category="/testing" command_run="echo 'hello world'" de
  name="ansibletest" action="ansibletest" category="/" desc="test ansible" playbook="playbooks/u
outbit> help
...
....
testing helloworld          print hello world
outbit> actions edit name=helloworld action=helloworld category=/ command_run="echo 'hello world'" de
outbit> help
...
....
helloworld          print hello world
outbit> actions del name=helloworld
deleted action pwd
```

2.2 Inventory

The inventory command is used to manage the outbit host inventory. The inventory is automatically discovered when you run an action that uses the ansible plugin. Changes are logged and can be checked using the “logs” command with the type=changes option.

```
outbit> inventory list
hostname1
outbit> inventory del name=hostname1
deleted inventory item hostname1
```

2.3 Jobs

The jobs command is used to manage running and previously completed jobs. A job is when you execute a specific action that is long running. The “jobs list” command will print the list of recently run jobs. The “jobs status” command

will show the result of the job.

This example assumes an action named `ansibletest` exists and uses the `ansible` plugin. Below is an example for creating the `ansibletest` action.

```
outbit> actions add name=ansibletest action=ansibletest category=/ desc="test ansible" playbook="update"
```

Below shows running the action which creates a job. `ctrl-z` is then pressed to background the job and the “jobs list” command is used to see the running job and previously run jobs. The “job status” command is used to check the result of the job.

```
outbit> ansibletest

Job is running with id=17. Press ctrl-z to background job.

outbit> jobs list
  Job ID      Is Running?   User      Command
  16          False         superadmin /ansibletest
  17          True          superadmin /ansibletest

outbit> jobs status id=17
Cloning into '/tmp/outbit/1467473131.22'...

outbit> jobs kill id=17
The job 17, was terminated
```

2.4 Logs

The `logs` command will display the history of everything run by outbit. Running `logs` without any options will result in showing all the requests made to the outbit api server, even invalid requests that could not be processed. If you specify a specific type, such as `type=changes`, it will show all the changes that were made to the outbit inventory. The option `type=changes` is useful to audit changes made to systems using outbit.

```
outbit> logs
  username      category      action  options  date
  superadmin    /users        list    None     06/18/2016 14:20
  superadmin    /              help    None     06/18/2016 14:20
  superadmin    /testing      ls       None     06/18/2016 14:20
  superadmin    /testing      pwd      None     06/18/2016 14:20
outbit> logs type=changes
  inventory_item  desc          job_id      date
  hostname1      setup         82          07/14/2016 35:04
  hostname1      apt-get update 82          07/14/2016 35:04
  hostname1      apt-get upgrade 82          07/14/2016 35:04
  hostname2      apt-get upgrade 84          07/14/2016 44:04
  hostname3      apt-get upgrade 83          07/14/2016 44:04
  hostname4      apt-get upgrade 85          07/14/2016 04:05
outbit> logs type=changes name=hostname4
  inventory_item  desc          job_id      date
  hostname4      apt-get upgrade 85          07/14/2016 04:05
```

2.5 Plugins

The `plugins` command will show plugins that are available to use in user defined actions.

```
outbit> plugins list
ansible
actions_list
roles_del
secrets_edit
logs
actions_add
schedules_add
secrets_add
jobs_kill
users_del
roles_list
inventory_list
users_edit
schedules_edit
users_add
secrets_del
help
roles_edit
schedules_list
actions_edit
inventory_del
users_list
roles_add
ping
actions_del
jobs_list
command
jobs_status
schedules_del
secrets_encryptpw
secrets_list
plugins_list
```

2.6 Roles

The roles command is used to manage Role Base Access control. With the roles command you can limit what actions and secrets users have access to. By default, when a user is created they do not have access to anything within outbit. By creating and assigning users to roles you can extend their permissions and ability to perform actions within outbit. On a default install of outbit the “super” role is created, this role has access to all actions “/” and by default only the superadmin is assigned to this role.

```
outbit> roles list
  actions="/"  name="super"  users="superadmin"
outbit> roles add name=auditor action="/logs" users="jdoe1,jdoe2"
  created role auditor
outbit> roles edit name=auditor action="/logs" users="jdoe1" secrets="secret1"
  modified role auditor
outbit> roles del name=auditor
  deleted role auditor
```

2.7 Schedules

The schedules command is used to manage scheduled jobs, it works similar to a linux cron.

```
outbit> schedules add name=patchdmz hour=16
  created schedule patchdmz
outbit> schedules edit name=patchdmz category=/ action=patchdmz minute=30 hour=20 day_of_month=* month=*
  modified schedule patchdmz
outbit> schedules list
  patchdmz
outbit> schedules del name=patchdmz
  deleted schedule patchdmz
```

2.8 Secrets

The `secrets` command is used to manage passwords, ssh keys, git keys, credentials, and other sensitive information that might be required by an action. These values are not visible to users after they are stored in the database, but actions users can execute may access them.

Adding, Editing, and Deleting Secrets .. sourcecode:: bash

```
outbit> secrets add name=neptune_rootpw secret='verysecretpassword' desc='root pw for neptune server'
  created secret neptune_rootpw
outbit> secrets edit name=neptune_rootpw secret='newpassword' desc='root pw for neptune server'
  modified secret neptune_rootpw
outbit> secrets list desc='root pw for neptune server' name='neptune_rootpw' secret='...' status='encrypted'
outbit> secrets del name=neptune_rootpw  deleted secret neptune_rootpw
```

Examples of migrating secrets encrypted with an older “`encryption_password`” or unencrypted secrets .. sourcecode:: bash

```
outbit> secrets encryptpw secret secretusinggoldpw encrypted using new password
outbit> secrets encryptpw name=noencsecret secret noencsecret updated to new password
```

2.9 Users

The `users` command is used to manage user accounts that can login and interact with outbit.

```
outbit> users list
  superadmin
outbit> users add username=jdoel password=jdoel
  created user jdoel
outbit> users edit username=jdoel password="new password"
  modified user jdoel
outbit> users del username=jdoel
  deleted user jdoel
```

To edit your own password, you can use “`users edit password=newpassword`”. In the below example the logged in user is `jdoel`. If the username is omitted then it changes the password of the current user.

```
outbit> users edit password="new password"
  modified user jdoel
outbit> users edit username=jdoel password="new password"
  modified user jdoel
```

3.1 GUI

The outbit GUI is best run using Docker.

Install and Starting the outbit api server, gui, and cli using Docker.

```
$ docker pull starboarder2001/outbit  
$ docker run -d -p 8088:8088 -p 80:80 -p 443:443 starboarder2001/outbit
```

Connect using the cli with the below command and make sure to change the default password of 'superadmin'.

Connect to the host on port 443 for the GUI.

outbit

outbit provides a simple UI for orchestrating changes or applying configurations in a datacenter and cloud environment. outbit provides a layer on top of Ansible that allows you to easily wrap up automated tasks and provide a simple way to execute them. The role based access control allows you to implement separations of duties and limit specific actions to be executed by specific roles. The logging feature allows you to track the history of changes in the environment.

Installation

Install outbit client only. This is if you already have a dedicated outbit api server.

```
$ pip install outbit
```

Install outbit api server.

```
$ pip install outbit  
$ sudo outbit-api-install
```

Install and Starting the outbit api server using Docker.

```
$ docker pull starboarder2001/outbit  
$ docker run -d -p 8088:8088 -p 80:80 -p 443:443 starboarder2001/outbit
```

Usage

Start the API server on your localhost or on a dedicated IP. If your using the Docker container then make sure you have pulled the image and have run the image using the above example.

```
$ outbit-api -s 127.0.0.1 --insecure
```

Login to the outbit shell. On the first login you will be prompted to change the default password. If your using the Docker container you can remove the “-insecure” flag since by default its configured to use ssl.

```
$ outbit -u superadmin -s 127.0.0.1 --insecure --no-check-certificates
Password: superadmin
Changing Password From Default
Enter New Password: *****
Enter New Password Again: *****
```

Example of adding a “hello world” action that prints hello world.

```
outbit> help
actions list          list actions
actions del          del actions
actions edit         edit actions
actions add          add actions
users list           list users
users del            del users
users edit           edit users
users add            add users
roles list           list roles
roles del            del roles
roles edit           edit roles
roles add            add roles
secrets list         list secrets
secrets del          del secrets
secrets edit         edit secrets
secrets add          add secrets
secrets encryptpw    Change password encryption
plugins list         list plugins
ping                 verify connectivity
logs                 show the history log
help                 print usage
jobs list            list jobs
jobs status          get status of job
jobs kill            kill a job
schedules add        add schedule
schedules edit       edit schedule
```

```
schedules list          list schedules
schedules del          del schedule
inventory list         list inventory
inventory del          del inventory item
exit

outbit> actions add name=helloworld category=/hello action=world plugin=command desc="print hello wor

outbit> help
actions list          list actions
actions del          del actions
actions edit         edit actions
actions add          add actions
users list           list users
users del            del users
users edit           edit users
users add            add users
roles list           list roles
roles del            del roles
roles edit           edit roles
roles add            add roles
secrets list         list secrets
secrets del          del secrets
secrets edit         edit secrets
secrets add          add secrets
secrets encryptpw    Change password encryption
plugins list         list plugins
ping                 verify connectivity
logs                 show the history log
help                 print usage
jobs list            list jobs
jobs status          get status of job
jobs kill            kill a job
schedules add        add schedule
schedules edit       edit schedule
schedules list       list schedules
schedules del        del schedule
inventory list       list inventory
inventory del        del inventory item
hello world          print hello world
exit

outbit> hello world
hello world
return code: 0

outbit> exit
```

License

outbit is released under the MIT License

Author

David Whiteside (david@dauidwhiteside.com)